

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Нижегородский государственный университет им. Н.И. Лобачевского

С.А. Белов, Н.Ю. Золотых

ЧИСЛЕННЫЕ МЕТОДЫ ЛИНЕЙНОЙ АЛГЕБРЫ

Лабораторный практикум

Нижний Новгород
Издательство Нижегородского госуниверситета
2005

УДК 519.61
ББК 22.193.1
Б435

Белов С.А., Золотых Н.Ю. Численные методы линейной алгебры. Лабораторный практикум. — Нижний Новгород: Изд-во Нижегородского госуниверситета им. Н.И. Лобачевского, 2005. — 264 с.

ISBN 5–85746–837–X

Учебное пособие посвящено численным методам линейной алгебры и затрагивает следующие темы: матричные нормы и число обусловленности, прямые и итерационные методы решения линейных систем, задачу наименьших квадратов, технологию разреженных матриц. По каждой теме даются необходимый теоретический материал и задания к лабораторным работам. Помимо этого, книга содержит руководство по учебно-практической библиотеке численных методов NL, доступный путеводитель по пакету LAPACK и его версии на языке C CLAPACK и краткое руководство по использованию системы MATLAB.

*Подготовлен в учебно-исследовательской лаборатории
«Математические и программные технологии
для современных компьютерных систем
(Информационные технологии)» факультета ВМК ННГУ
при поддержке Фонда содействия развитию малых
форм предприятий в научно-технической сфере.*

ISBN 5–85746–837–X

ББК 22.193.1

© С.А. Белов, Н.Ю. Золотых, 2005

Оглавление

Предисловие	7
Обозначения и соглашения	10
Глава 1. Машинная арифметика и анализ ошибок	12
1.1. Нормы векторов и матриц	12
1.2. Число обусловленности матрицы	14
1.2.1. Оценки для числа обусловленности	15
1.2.2. Решение «возмущенной» системы	17
1.2.3. Апостериорные оценки	18
1.2.4. Покомпонентная относительная обратная ошибка	18
1.3. Машинная арифметика	19
1.3.1. Числа с плавающей точкой	19
1.3.2. IEEE-арифметика	22
1.4. Прямой и обратный анализ ошибок	25
1.4.1. Прямой анализ ошибок	25
1.4.2. Анализ чувствительности	26
1.4.3. Обратный анализ ошибок	27
1.5. Производительность алгоритмов	28
1.6. Библиотека NL	28
Глава 2. Прямые методы решения линейных систем	33
2.1. Введение	33
2.2. Метод Гаусса	33
2.3. LU -разложение	36
2.3.1. Исключение по столбцу	37
2.3.2. Исключение по строке	38
2.3.3. Компактная схема	39
2.3.4. Замечания о производительности	41
2.4. Схемы выбора ведущего элемента	42
2.4.1. Стратегия частичного выбора	42

2.4.2.	Стратегия полного выбора	44
2.5.	Анализ ошибок	44
2.5.1.	Анализ ошибок в LU -разложении	45
2.5.2.	Анализ ошибок при решении треугольных систем	46
2.5.3.	Рост коэффициентов	49
2.6.	Оценка числа обусловленности	51
2.7.	Итерационное уточнение	52
2.8.	Масштабирование	54
2.9.	LU -разложение в библиотеке NL	54
2.10.	LDL^T -разложение	57
2.11.	Разложение Холецкого	58
2.12.	Разложение Холецкого в библиотеке NL	61
Глава 3.	Линейная задача наименьших квадратов	64
3.1.	Постановка задачи	64
3.2.	Обусловленность задачи наименьших квадратов	66
3.3.	Нормальные уравнения	67
3.4.	QR -разложение	69
3.4.1.	Ортогонализация Грама–Шмидта	70
3.4.2.	Отражения Хаусхолдера	72
3.4.3.	Вращения Гивенса	75
3.5.	QR -разложение в библиотеке NL	78
3.6.	Сингулярное разложение (SVD)	83
3.7.	Сравнение методов	85
3.8.	Задача наименьших квадратов неполного ранга	86
3.8.1.	Использование сингулярного разложения	87
3.8.2.	Псевдообратная матрица	89
3.9.	Сингулярное разложение в библиотеке NL	90
Глава 4.	Разреженные системы	94
4.1.	Введение	94
4.2.	Способы представления разреженных матриц	94
4.2.1.	Разреженный строчный формат	95
4.2.2.	Разреженный столбцовый формат	97
4.2.3.	Формат для хранения разреженных симметричных матриц	98
4.3.	Основные операции с разреженными матрицами	99
4.4.	Разреженные матрицы в библиотеке NL	99

4.5. Разреженные системы линейных уравнений	102
4.6. Разреженные системы в библиотеке NL	106
4.7. Ленточные матрицы	109
4.7.1. Трехдиагональные системы	109
4.7.2. Метод прогонки в библиотеке NL	111
4.7.3. Ленточные системы	112
4.7.4. Ленточные системы в библиотеке NL	114
Глава 5. Итерационные методы	118
5.1. Метод простой итерации	119
5.2. Метод Якоби	121
5.3. Метод Гаусса–Зейделя	122
5.4. Метод релаксации	122
5.5. Метод релаксации в библиотеке NL	123
5.6. Метод сопряженных градиентов	126
5.7. Метод сопряженных градиентов в библиотеке NL	131
5.8. Метод бисопряженных градиентов	134
5.9. Метод бисопряженных градиентов в библиотеке NL	136
5.10. Метод минимальных невязок	139
5.11. Метод наискорейшего спуска	140
Глава 6. Использование системы МАТЛАВ	141
6.1. Краткое введение в МАТЛАВ	142
6.2. Основные функции для работы с матрицами	146
6.3. Системы линейных уравнений	156
6.3.1. Решатели систем линейных уравнений	156
6.3.2. LU -разложение	159
6.3.3. Разложение Холецкого	160
6.3.4. QR -разложение	161
6.3.5. Сингулярное разложение	163
6.3.6. Нормы векторов и матриц	164
6.3.7. Число обусловленности	165
6.3.8. Оценка числа обусловленности	165
6.3.9. Обратная и псевдообратная матрицы	167
6.4. Разреженные матрицы	169
6.4.1. Создание и преобразование разреженных матриц	169
6.4.2. Алгоритмы переупорядочения	172
6.4.3. Неполная факторизация	176

6.4.4. Решение систем линейных уравнений	179
Глава 7. Библиотека LAPACK	184
7.1. Обзор библиотеки	184
7.2. Решение систем линейных уравнений	188
7.2.1. Функция <i>DGESV</i>	188
7.2.2. Функция <i>DGESVX</i>	192
7.3. Решение положительно определенных систем	202
7.4. Решение трехдиагональной системы	206
7.5. Решение ленточных систем	210
7.6. Линейная задача наименьших квадратов	216
7.6.1. Использование <i>QR</i> - и <i>QL</i> -разложений	216
7.6.2. Использование сингулярного разложения	223
7.7. Рекомендации по установке библиотеки	228
Глава 8. Библиотека SLAPACK	230
8.1. Рекомендации по установке библиотеки	230
8.2. Особенности вызова функций	232
8.3. Пример	235
Глава 9. Задачи	238
9.1. Задачи к первой главе	238
9.2. Задачи ко второй главе	243
9.3. Задачи к третьей главе	249
9.4. Задачи к четвертой главе	254
9.5. Задачи к пятой главе	256
Литература	257
Предметный указатель	259

Предисловие

Учебное пособие, предлагаемое вниманию читателя, посвящено численным методам линейной алгебры. В нем затрагиваются следующие темы: матричные нормы и число обусловленности, прямые и итерационные методы решения линейных систем, задача наименьших квадратов, технология разреженных матриц. По каждой теме мы даем необходимый теоретический материал и задания к лабораторным работам. Помимо этого книга содержит руководство по учебно-практической библиотеке численных методов NL, доступный путеводитель по пакету LAPACK и его версии на языке C CLAPACK и краткое руководство по использованию системы МАТЛАВ. Мы полагаем, что при выполнении задания студент воспользуется указанным программным обеспечением или, возможно, напишет свою программу «с нуля».

На сегодняшний день объем информации в области методов вычислений практически необозрим. Существуют большие библиотеки программ, такие, как библиотека Netlib [32] (включающая пакеты LINPACK, EISPACK, LAPACK, BLAS и др.), библиотеки численного анализа Numerical Algorithms Group — NAG [31], научная библиотека GSL ассоциации GNU [28], библиотека НИВЦ МГУ [24] и др. Указанные библиотеки широко применяются современными вычислителями, однако в учебных курсах и практикумах используются достаточно редко. К причинам последнего (среди прочих) следует отнести некоторую их «громоздкость». Достаточно сложны соглашения о параметрах функций. Исходный код, как правило, написан на Фортране. Эти свойства часто затрудняют возможность использования исходного кода этих пакетов в рамках студенческих

лабораторных работ. Поэтому мы не справились с искушением написать свою собственную библиотеку. Предлагаемая библиотека NL написана А.М. Тагуновым и Н.Ю. Золотых и протестирована стажерами лаборатории «Информационные технологии» А.Н. Половинкиным, К.В. Корняковым и В.В. Рябовым. Она, конечно, уступает по производительности таким современным пакетам, как LAPACK, однако ее исходный код намного более прозрачен и студент всегда может обратиться к нему для уяснения деталей алгоритма или с целью внесения изменений.

Для работы с пособием необходимы знания линейной алгебры и математического анализа в объеме соответствующих университетских курсов для студентов естественно-научных специальностей. Предполагается, что лабораторные работы выполняются параллельно с изучением теоретического материала по численным методам. В качестве учебников по современным численным методам линейной алгебры мы рекомендуем [2, 4, 5], в качестве справочника — [3]. При написании пособия мы сами активно пользовались этими книгами. Описание функций и подпрограмм в главах 6, 7 следует руководствам [18, 23]. По духу наше пособие близко учебникам [1, 10, 17, 21, 22].

В книге приводятся доказательства только некоторых ключевых утверждений, что вполне, как мы считаем, оправдывается поставленными задачами. Опущенные доказательства во многих случаях могут быть восстановлены: это, по мнению авторов, являлось бы хорошим упражнением. В тех случаях, когда известные нам доказательства сложны, мы всегда приводим ссылку на соответствующую литературу.

Мы благодарим за сотрудничество директора по развитию Нижегородской лаборатории корпорации «Интел» Л.В. Нестеренко. Во время работы над пособием авторы получали финансовую поддержку Фонда содействия развитию малых форм предприятий в научно-технической сфере. Мы искренне при-

знательны его генеральному директору И.М. Бортнику и начальнику отдела развития инфраструктуры В.Н. Оганесяну.

Мы надеемся, что настоящее пособие будет полезно студентам и всем тем, кто на практике применяет численные методы линейной алгебры.

Обозначения и соглашения

В пособии рассматриваются задачи линейной алгебры над полем вещественных чисел, поэтому все скаляры, векторы и матрицы считаются вещественными (не комплексными). Заметим, что некоторые приведенные результаты и описанные алгоритмы без труда переносятся и на комплексный случай, тогда как для некоторых алгоритмов и фактов такое перенесение невозможно.

Под арифметическим вещественным пространством понимается линейное пространство вещественных столбцов заданной высоты со стандартно определенными операциями сложения и умножения вектора на вещественный скаляр.

Как правило, векторы мы обозначаем малыми латинскими буквами (a, b, c, \dots), матрицы — большими латинскими буквами (A, B, C, \dots), а их элементы соответствующими малыми латинскими буквами с индексами ($a_j, b_j, c_j, \dots, a_{ij}, b_{ij}, c_{ij}, \dots$). Если матрица задана выражением, например $A + B$, то для ссылки на ее элемент i -й строки j -го столбца будем писать $(A + B)_{ij}$. Аналогичные соглашения примем и для векторов. Под неравенством $A \leq B$ понимается система покомпонентных неравенств $a_{ij} \leq b_{ij}$ для всех i и j .

E единичная матрица (порядок ясен из контекста)

e_j j -й столбец единичной матрицы

$\|x\|, \|x\|_p$ норма вектора x ; см. стр. 12

$\|A\|, \|A\|_p$ норма матрицы A ; см. стр. 13

$\text{rank } A$ ранг матрицы A

$\det A$ определитель матрицы A

A^T	транспонированная матрица
$ A $	матрица $B = (b_{ij})$, в которой $b_{ij} = a_{ij} $
$L(a_1, \dots, a_n)$	линейная оболочка векторов a_1, \dots, a_n
$\text{diag}(\lambda_1, \dots, \lambda_n)$	диагональная матрица с элементами $\lambda_1, \dots, \lambda_n$ на диагонали
$\text{diag}(D_1, \dots, D_k)$	блочно-диагональная матрица с блоками D_1, \dots, D_k на диагонали
$\lambda_1(A), \dots, \lambda_n(A)$	собственные числа квадратной матрицы A порядка n
$\sigma_1(A), \dots, \sigma_n(A)$	сингулярные числа матрицы A размера $m \times n$ ($m \geq n$): $\sigma_j(A) = \sqrt{\lambda_j(A^T A)}$
$\text{cond } A, \text{cond}_p A$	число обусловленности матрицы A ; см. стр. 14, 66
$\text{cond}_{LS} A$	число обусловленности матрицы A для задачи наименьших квадратов; см. стр. 67
$\rho(A)$	спектральный радиус матрицы A ; см. стр. 119
$\text{fl}(x)$	представление в компьютере числа x ; см. стр. 20
ε_M	машинное эpsilon; см. стр. 21
$\text{sign}(\alpha)$	знак числа α ; равен 1, 0 или -1 , если соответственно $\alpha > 0$, $\alpha = 0$, $\alpha < 0$
g_{pp}, g_{cp}	коэффициенты роста элементов в алгоритме LU -разложения с частичным и соответственно полным выбором ведущего элемента; см. стр. 49, 50

Глава 3

Линейная задача наименьших квадратов

3.1. Постановка задачи

Пусть A — матрица размера $m \times n$, а b — столбец высоты m . *Линейная задача наименьших квадратов* заключается в нахождении такого вектора x , называемого *псевдорешением* системы $Ax = b$, на котором достигается минимум

$$\min_x \|Ax - b\|_2.$$

Таким образом, псевдорешение доставляет минимум евклидовой нормы невязки заданной системы $Ax = b$. Легко видеть, что если система $Ax = b$ совместна, то любое ее решение является ее псевдорешением. Если $m > n$, то система называется *переопределенной*.

Псевдорешение единственно тогда и только тогда, когда $\text{rang } A = n$ (столбцы матрицы A линейно независимы).

Задача наименьших квадратов возникает при *аппроксимации* данных. Предположим, что имеются пары значений

$$(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m). \quad (33)$$

Пусть $f_1(x), f_2(x), \dots, f_n(x)$ — заданные функции. Требуется найти функцию вида

$$f(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \dots + \alpha_n f_n(x), \quad (34)$$

для которой $f(x_i) \approx y_i$ ($i = 1, 2, \dots, m$), т. е. функцию, приближающую данные (33). Более точно: требуется найти значения параметров $\alpha_1, \dots, \alpha_n$, на которых достигается минимум

$$\min_{\alpha_1, \dots, \alpha_n} \sum_{i=1}^m (y_i - \alpha_1 f_1(x_i) - \dots - \alpha_n f_n(x_i))^2.$$

Легко видеть, что набор этих параметров является псевдо-решением системы

[illegible]

И наоборот, если $\alpha_1, \dots, \alpha_n$ — псевдорешение указанной системы, то функция $f(x)$, определяемая по формуле (34), является решением задачи аппроксимации.

Часто в качестве функций $f_1(x), f_2(x), \dots, f_n(x)$ рассматриваются многочлены $1, x, \dots, x^{n-1}$. Таким образом, речь идет об аппроксимации данных полиномами.

Мы рассмотрим следующие методы решения линейной задачи наименьших квадратов:

- 1) нормальные уравнения;
- 2) QR -разложение:
 - а) ортогонализация Грама–Шмидта,
 - б) отражения Хаусхолдера,
 - в) вращения Гивенса,
- 3) сингулярное разложение (SVD).

Далее мы, как правило, сосредоточимся на случае полного ранга ($\text{rank } A = n$). В разделе 3.8 рассматривается (более трудная) задача наименьших квадратов неполного ранга.

Подробно линейная задача наименьших квадратов рассматривается, например, в [4, 5, 13].

3.2. Обусловленность задачи наименьших квадратов

В разделе 1.2 было введено число обусловленности квадратной матрицы. Дадим определение числа обусловленности прямоугольной матрицы. Пусть матрица A имеет размеры $m \times n$, причем $m \geq n$. *Спектральным числом обусловленности матрицы A* называется величина

$$\text{cond}_2 A = \frac{\max_j \sigma_j(A)}{\min_j \sigma_j(A)}.$$

Это определение согласуется с определением $\text{cond}_2 A$ для квадратной матрицы (см. (2) на стр. 14). Если $m < n$, то по определению $\text{cond}_2 A = \text{cond}_2 A^T$.

Предположим, что $\text{rank } A = n$. Пусть x — решение задачи наименьших квадратов $\min \|Ax - b\|_2$, r — невязка, т.е. $r = b - Ax$, $x + \Delta x$ — решение «возмущенной» задачи наименьших квадратов $\min \|(A + \Delta A)x - (b + \Delta b)\|_2$. Предположим, что возмущение не слишком велико, а именно, $\varepsilon < 1/\text{cond}_2 A$, где

$$\varepsilon = \max \left\{ \frac{\|\Delta A\|_2}{\|A\|_2}, \frac{\|\Delta b\|_2}{\|b\|_2} \right\}.$$

Можно показать [4] (см. также задачу 3 из раздела 9.3 на стр. 250), что

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leq \varepsilon \cdot \left(\frac{2 \text{cond}_2 A}{\cos \theta} + \text{tg } \theta \cdot \text{cond}_2^2 A \right) + O(\varepsilon^2), \quad (35)$$

где $\sin \theta = \|r\|_2/\|b\|_2$ (таким образом, θ — это угол между векторами b и Ax).

Величина

$$\text{cond}_{\text{LS}}(A, b) = \frac{2 \text{cond}_2 A}{\cos \theta} + \text{tg } \theta \cdot \text{cond}_2^2 A$$

называется *числом обусловленности для задачи наименьших квадратов*. Получаем, что

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leq \varepsilon \text{cond}_{\text{LS}}(A, b) + O(\varepsilon^2).$$

Итак, $\text{cond}_{\text{LS}}(A, b)$ является мерой чувствительности задачи наименьших квадратов, показывающей, как сильно решение этой задачи зависит от возмущения входных данных.

Формулу (35) можно интерпретировать следующим образом. Если угол θ мал, то $\text{cond}_{\text{LS}}(A, b) \approx 2 \text{cond}_2 A$, т. е. задача наименьших квадратов обусловлена примерно так же, как и задача решения систем линейных уравнений.

Если угол θ не мал, но и не близок к $\pi/2$, то $\text{cond}_{\text{LS}}(A, b) \approx \text{cond}_2^2 A$.

Если $\theta \approx \pi/2$, т. е. задача близка к вырожденной, то величина $\text{cond}_{\text{LS}}(A, b)$ растет неограничено, даже если $\text{cond}_2^2 A$ имеет умеренные значения.

3.3. Нормальные уравнения

Покажем, что множество псевдорешений системы $Ax = b$ совпадает с множеством решений системы

$$A^T Ax = A^T b, \quad (36)$$

называемой системой *нормальных уравнений*.

Имеем

$$\begin{aligned} \|A(x + \Delta x) - b\|_2^2 &= (A(x + \Delta x) - b)^T (A(x + \Delta x) - b) = \\ &= (A\Delta x)^T (A\Delta x) + (Ax - b)^T (Ax - b) + 2(A\Delta x)^T (Ax - b) = \\ &= \|A\Delta x\|_2^2 + \|Ax - b\|_2^2 + 2\Delta x^T (A^T Ax - A^T b). \end{aligned}$$

Если x — псевдорешение, то $\|A\Delta x\|_2^2 + 2\Delta x^T(A^T Ax - A^T b) \geq 0$ для любого Δx , откуда $A^T Ax - A^T b = 0$. Обратно, если $A^T Ax = A^T b$, то $\|A(x + \Delta x) - b\|_2^2 \geq \|Ax - b\|_2^2$ для любого Δx , т. е. x — псевдорешение.

Заметим, что матрица $A^T A$ — квадратная и неотрицательно определенная. Система (36) всегда совместна. Она имеет единственное решение тогда и только тогда, когда псевдорешение исходной системы единственно. Последнее имеет место в том и только в том случае, когда столбцы матрицы A линейно независимы (т. е. $\text{rank } A = n$). В этом (и только в этом) случае матрица $A^T A$ положительно определена и для решения системы (36) мы можем применить разложение Холецкого. Тогда на вычисление матричного произведения $A^T A$ требуется $mn^2 + O(n^2)$ операций с плавающей точкой (достаточно вычислить только элементы над диагональю), а на решение системы (36) требуется $n^3/3 + O(n^2)$ операций с плавающей точкой. Так как $m \geq n$, то основная масса времени тратится на формирование матрицы $A^T A$.

Недостатком метода нормальных уравнений является плохая обусловленность получаемой системы. Действительно, справедливо равенство $\text{cond}_2(A^T A) = \text{cond}_2^2 A$. Используя результаты анализа ошибок для метода Холецкого, приходим к выводу, что относительная ошибка получаемого решения задачи наименьших квадратов не превосходит некоторого умеренного кратного величины $\varepsilon_M \text{cond}_2^2 A$. Следует ожидать, что при решении нормальных уравнений с небольшой невязкой исходной системы будет потеряно вдвое больше разрядов, чем при использовании QR -разложения или сингулярного разложения.

Тем не менее использование нормальных уравнений заманчиво при решении линейной задачи наименьших квадратов при малых значениях n : например, при $n = 2$ система (36) имеет размеры 2×2 .

3.4. QR-разложение

Можно доказать, что для произвольной $(m \times n)$ -матрицы A ранга n существуют и единственны матрицы Q размера $m \times n$ с ортонормированными столбцами (т.е. $Q^T Q = E$) и верхнетреугольная матрица R размера $n \times n$ с положительными диагональными элементами, такие, что

$$A = QR. \quad (37)$$

Разложение (37) называется *QR-разложением*¹ матрицы A .

Сперва покажем, как решать линейную задачу наименьших квадратов $\min \|Ax - b\|_2$, если QR-разложение матрицы A известно. Методы построения самого QR-разложения будут рассмотрены позднее.

Итак, $Q^T Q = E$. Дополним столбцы матрицы Q до ортонормированной системы m векторов². Из построенных столбцов составим матрицу S . Имеем $S^T Q = 0$. Матрица (Q, S) — ортогональная, т.е. $(Q, S)^{-1} = (Q, S)^T$, поэтому для любого вектора x справедливо $\|(Q, S)^T x\|_2 = \|x\|_2$. Теперь получаем:

$$\|Ax - b\|_2^2 = \|(Q, S)^T (QRx - b)\|_2^2 = \left\| \begin{pmatrix} Q^T QRx - Q^T b \\ S^T QRx - S^T b \end{pmatrix} \right\|_2^2 =$$

¹Иногда QR-разложением называют представление матрицы A в виде произведения $A = \tilde{Q}\tilde{R}$, в котором матрица \tilde{Q} — квадратная ортогональная порядка m , а матрица \tilde{R} — верхнетреугольная размера $m \times n$. Чтобы такое разложение получить по разложению (37), достаточно столбцы матрицы Q дополнить до ортонормированной системы m векторов, из построенных столбцов составить матрицу S и положить

$$\tilde{Q} = (Q, S), \quad \tilde{R} = \begin{pmatrix} R \\ 0 \end{pmatrix}.$$

Обратный переход осуществляется отбрасыванием в \tilde{Q} последних $m - n$ столбцов, а в \tilde{R} — последних $m - n$ строк.

²Ср. предыдущее примечание.

$$= \|Rx - Q^T b\|_2^2 + \|S^T b\|_2^2.$$

Минимум достигается тогда и только тогда, когда первое слагаемое равно нулю. Итак, решением линейной задачи наименьших квадратов является решение треугольной системы

$$Rx = Q^T b,$$

при этом

$$\min_x \|Ax - b\|_2 = \|S^T b\|_2.$$

Заметим, что матрица R невырождена тогда и только тогда, когда $\text{rank } A = n$. В этом случае $x = R^{-1}Q^T b$.

Рассмотрим несколько методов построения QR -разложения.

3.4.1. Ортогонализация Грама–Шмидта

Процесс ортогонализации Грама–Шмидта — это классический метод построения QR -разложения. Он заключается в построении ортонормированного базиса q_1, \dots, q_n подпространства, натянутого на столбцы a_1, \dots, a_n , причем

$$L(q_1, \dots, q_j) = L(a_1, \dots, a_j) \quad (j = 1, 2, \dots, n). \quad (38)$$

Матрица, составленная из столбцов q_1, \dots, q_n , — это матрица Q . Матрица перехода от базиса q_1, \dots, q_n к a_1, \dots, a_n — это матрица R , т. е. $a_j = \sum_{i=1}^j r_{ij} q_i$

Опишем этот алгоритм подробнее.

for $j = 1, \dots, n$

$q_j = a_j$

for $i = 1, \dots, j - 1$

$r_{ij} = q_i^T a_j$

$q_j = q_j - r_{ij} q_i$

end

$r_{jj} = \|q_j\|_2$

(*)

```

if  $r_{jj} = 0$ 
    exit ( $a_i$  линейно зависит от  $a_1, \dots, a_{i-1}$ )
end
 $q_j = q_j / r_{jj}$ 
end

```

Алгоритм использует $2mn^2 + O(mn)$ операций с плавающей точкой.

Заметим, что присваивание $r_{ij} = q_i^T a_j$ в строке (*) приведенного алгоритма можно заменить на $r_{ij} = q_i^T q_j$. Полученный таким образом новый алгоритм (в точной арифметике эквивалентный исходному) называется *модифицированным процессом Грама–Шмидта*.

При реальных вычислениях на компьютере (при наличии ошибок округления) процесс ортогонализации Грама–Шмидта (в том числе модифицированный) вместо векторов q_1, q_2, \dots, q_n возвратит векторы $q_1, q_2 + \Delta q_2, \dots, q_n + \Delta q_n$. Например, в режиме накопления

$$L(q_1, q_2 + \Delta q_2, \dots, q_j + \Delta q_j) = L(a_1, a_2 + \Delta a_2, \dots, a_j + \Delta a_j),$$

где $\|\Delta a_k\|_2 \leq 2\varepsilon_M \|a_k\|_2$, т.е. линейные оболочки полученных векторов совпадают с линейными оболочками «слабо возмущенных» исходных векторов [2].

Итак, в отношении выполнения равенства (38) процесс ортогонализации Грама–Шмидта ведет себя крайне устойчиво. К сожалению, этого нельзя сказать о выполнении условия ортогональности полученной системы. В большинстве случаев процесс Грама–Шмидта приводит к значительному нарушению ортогональности. Модифицированный процесс Грама–Шмидта чуть лучше, но и он не свободен от указанного недостатка. Пусть $Q + \Delta Q$ — реально вычисленная матрица. Можно показать [4], что

$$(Q + \Delta Q)^T \cdot (Q + \Delta Q) = E + \Delta E,$$

где $\|\Delta E\|_2 \lesssim \varepsilon_M \operatorname{cond}_2 A$.

Для исправления отмеченного недостатка используют *пертортогонализацию* [2].

Тем не менее, как оказалось, ортогональность столбцов матрицы Q не играет решающей роли в задаче наименьших квадратов. Можно показать, что алгоритм решения задачи наименьших квадратов с помощью модифицированного процесса ортогонализации Грама–Шмидта обратно устойчив и, следовательно может быть использован на практике [4]. В любом случае, затраты в методах типа Грама–Шмидта чуть выше, чем в методе Хаусхолдера.

3.4.2. Отражения Хаусхолдера

Метод отражений Хаусхолдера — один из самых распространенных методов нахождения QR -разложения.

Пусть v — n -мерный ненулевой вектор-столбец. Квадратная матрица P порядка n вида

$$P = E - \frac{2vv^T}{v^T v}$$

называется *отражением Хаусхолдера* или просто *отражением*. Вектор v называется *вектором Хаусхолдера*. Умножению матрицы P на вектор x можно дать следующую геометрическую интерпретацию: вектор Px получается отражением вектора x относительно гиперплоскости, ортогональной вектору v . Легко проверить, что матрица Хаусхолдера симметрична и ортогональна.

Пусть $x \neq 0$. Найдем отражение P , такое, что $Px = \alpha e_1$ для какого-либо α . Имеем

$$Px = \left(E - \frac{2vv^T}{v^T v} \right) x = x - \frac{2v(v^T x)}{v^T v} = x - \frac{2v^T x}{v^T v} v,$$

и поэтому v коллинеарен $x - Px = x - \alpha e_1$. Так как v определен с точностью до ненулевого множителя, то можно положить $v =$

$x - \alpha e_1$. Ввиду ортогональности матрицы P имеем $\|Px\|_2 = \|x\|_2$, откуда $\alpha = \pm\|x\|_2$. Поэтому

$$v = x \pm \|x\|_2 e_1. \quad (39)$$

Итак, в качестве вектора Хаусхолдера v можно взять вектор, вычисляемый по формуле (39), или любой ненулевой, ему коллинеарный.

С алгоритмом построения вектора Хаусхолдера связаны некоторые важные детали. Одна из них — выбор знака в формуле (39). Если x почти коллинеарен e_1 , то вектор $v = x - \text{sign}(x_1)\|x\|_2 e_1$ имеет малую норму. Вследствие этого возможно появление большой относительной ошибки при вычислении множителя $2/v^T v$. Эту трудность можно обойти, взяв α с тем же знаком, что и первая компонента вектора x :

$$v = x + \text{sign}(x_1)\|x\|_2 e_1. \quad (40)$$

Это обеспечивает выполнение неравенства $\|v\|_2 \geq \|x\|_2$ и гарантирует почти точную ортогональность вычисленной матрицы P .

Другая деталь связана с выбором множителя для вектора, вычисляемого по формуле (39). Мы будем использовать такой множитель, что $v_1 = 1$. Это несколько упрощает способ хранения ортогональных матриц, вычисленных по методу отражений Хаусхолдера.

Суть алгоритма QR-разложения на основе отражений сначала поясним на примере. Пусть $m = 5, n = 4$ и матрицы P_1 и P_2 таковы, что

$$P_2 P_1 A = \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ 0 & f_{22} & f_{23} & f_{24} \\ 0 & 0 & f_{33} & f_{34} \\ 0 & 0 & f_{43} & f_{44} \\ 0 & 0 & f_{53} & f_{54} \end{pmatrix}.$$

Покажем, как на месте элементов f_{43} , f_{53} получить нули. Найдем матрицу отражения P'_3 , такую, что

$$P'_3 \times \begin{pmatrix} f_{33} \\ f_{43} \\ f_{53} \end{pmatrix} = \begin{pmatrix} f'_{33} \\ 0 \\ 0 \end{pmatrix}.$$

Если $P_3 = \text{diag}(1, 1, P'_3)$, то

$$P_3 P_2 P_1 A = \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ 0 & f_{22} & f_{23} & f_{24} \\ 0 & 0 & f'_{33} & f'_{34} \\ 0 & 0 & 0 & f'_{44} \\ 0 & 0 & 0 & f'_{54} \end{pmatrix}.$$

Выполнив n таких шагов, получим верхнюю треугольную матрицу $\tilde{R} = P_n P_{n-1} \dots P_1 A$. Откуда, так как матрицы P_j — ортогональные и симметричные, имеем $A = \tilde{Q} \tilde{R}$, где $\tilde{Q} = P_1 P_2 \dots P_n$. Пусть Q состоит из первых n столбцов матрицы \tilde{Q} , а R состоит из первых n строк матрицы \tilde{R} , тогда $A = QR$. Это и есть искомое разложение.

Матрицы P_j не требуется формировать в явном виде: достаточно хранить соответствующий вектор v_j , который можно записывать в j -й столбец матрицы A .

Для решения задачи наименьших квадратов необходимо вычислить произведение $Q^T b = P_n P_{n-1} \dots P_1 b$.

Трудоемкость алгоритма построения QR -разложения на основе отражений составляет $2n^2 m - (2/3)n^3 + O(mn)$ арифметических операций с плавающей точкой. Для получения решения задачи наименьших квадратов при известном QR -разложении требуется только $O(mn)$ арифметических операций.

Пусть при реальных вычислениях на компьютере в результате ошибок округления вместо Q получена матрица $Q + \Delta Q$.

Можно показать [4], что

$$(Q + \Delta Q)^T \cdot (Q + \Delta Q) = E + \Delta E,$$

где $\|\Delta E\|_2 \lesssim \varepsilon_M$.

Можно показать [13], что при наличии ошибок округления вычисленное по методу Хаусхолдера решение $x + \Delta x$ задачи $\min \|Ax - b\|_2$ является точным решением «близкой» задачи

$$\min \|(A + \Delta A)x - (b + \Delta b)\|_2,$$

где

$$\begin{aligned} \|\Delta A\|_F &\leq (6m - 3n + 41)n\varepsilon_M \|A\|_F + O(\varepsilon_M^2), \\ \|\Delta b\|_2 &\leq (6m - 3n + 40)n\varepsilon_M \|b\|_2 + O(\varepsilon_M^2). \end{aligned} \quad (41)$$

Как мы видим, относительная ошибка эквивалентного возмущения пропорциональна умеренному кратному числу ε_M . Используя теорию возмущений из раздела 3.2, получаем, что при малой невязке относительная ошибка решения задачи наименьших квадратов примерно пропорциональна $\varepsilon_M \operatorname{cond}_2 A$, а при большой невязке — $\varepsilon_M \operatorname{cond}_2^2 A$.

На выходе диагональ и верхняя треугольная часть матрицы A содержат элементы матрицы R . Матрица Q представлена элементами ниже диагонали и вектором t . Вектор t имеет длину n . Матрица Q определяется произведением матриц отражений Хаусхолдера.

3.4.3. Вращения Гивенса

Вращением Гивенса, или просто *вращением*, называется матрица

$$R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}.$$

Умножению матрицы R на вектор x можно дать следующую геометрическую интерпретацию: вектор Rx получается вращением вектора x на угол θ . Вращение n -мерного пространства в плоскости x_i, x_j определяет матрица

$$R = \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & \cos \theta & & & -\sin \theta & \\ & & & & 1 & & & \\ & & & & & \ddots & & \\ & & & & & & 1 & \\ & & \sin \theta & & & & \cos \theta & \\ & & & & & & & 1 \\ & & & & & & & & \ddots & \\ & & & & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ j \\ \\ \\ i \\ \\ \\ \\ \end{matrix}$$

Очевидно, что матрица вращения ортогональна.

Если

$$\cos \theta = \frac{x_j}{\sqrt{x_i^2 + x_j^2}}, \quad \sin \theta = -\frac{x_i}{\sqrt{x_i^2 + x_j^2}},$$

то

$$\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \times \begin{pmatrix} x_j \\ x_i \end{pmatrix} = \begin{pmatrix} \sqrt{x_i^2 + x_j^2} \\ 0 \end{pmatrix}.$$

Таким образом, с помощью домножения слева вектора x на матрицу вращения R можно занулить одну (i -ю) компоненту. Заметим, что при этом все компоненты, кроме i -й и j -й, не изменяются.

Алгоритм QR -разложения на основе вращений аналогичен алгоритму, использующему отражения, с тем отличием, что

элементы матрицы A при обработке j -го столбца в нем зануляются по одному (для определенности, например, сверху вниз).

Чтобы перейти, например, от матрицы

$$\begin{pmatrix} f_{11} & f_{12} & f_{13} \\ 0 & f_{22} & f_{23} \\ 0 & f_{32} & f_{33} \\ 0 & f_{42} & f_{43} \end{pmatrix} \quad \text{к матрице} \quad \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ 0 & g_{22} & g_{23} \\ 0 & 0 & g_{33} \\ 0 & 0 & g_{43} \end{pmatrix},$$

производим умножения:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & -s & 0 \\ 0 & s & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ 0 & f_{22} & f_{23} \\ 0 & f_{32} & f_{33} \\ 0 & f_{42} & f_{43} \end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ 0 & f'_{22} & f'_{23} \\ 0 & 0 & f'_{33} \\ 0 & f_{42} & f_{43} \end{pmatrix},$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c' & 0 & -s' \\ 0 & 0 & 1 & 0 \\ 0 & s' & 0 & c' \end{pmatrix} \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ 0 & f'_{22} & f'_{23} \\ 0 & 0 & f'_{33} \\ 0 & f_{42} & f_{43} \end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ 0 & f''_{22} & f''_{23} \\ 0 & 0 & f'_{33} \\ 0 & 0 & f''_{43} \end{pmatrix}.$$

Для хранения элементов найденного разложения дополнительная память не нужна. Информация о матрице вращения, с помощью которой мы занулили соответствующий элемент, хранится на его месте.

Для этого применяют следующий метод [5]. Пусть $s = \sin \theta$, $c = \cos \theta$. Если $|s| < |c|$, то в память записывается число $s \cdot \text{sign}(c)$, в противном случае записывается $\text{sign}(s)/c$.

По хранимому значению p восстановление чисел s и c осуществляется следующим образом. Если $|p| < 1$, то $s = p$ и $c = \sqrt{1 - s^2}$, в противном случае $c = 1/p$ и $s = \sqrt{1 - c^2}$. Заметим, что вместо исходных s , c мы можем восстановить $-s$, $-c$, но для наших целей приемлема и такая пара.

Метод вращений в два полтора раза медленнее метода отражений. Существует метод *быстрых вращений* [4], который, все-таки, медленнее метода отражений.

Обратный анализ ошибок для метода Гивенса решения задачи наименьших квадратов приводит к оценкам, аналогичным (41) [4]. Относительная ошибка эквивалентного возмущения пропорциональна умеренному кратному числа ε_M .

3.5. QR-разложение в библиотеке NL

Метод QR-разложения на основе отражений Хаусхолдера реализован в функции *qr_decomp*. После того как разложение найдено, для решения квадратной совместной системы линейных уравнений необходимо воспользоваться функцией *qr_solve*. Приведем пример.

Листинг xqr.c

```
/*
Пример использования функций из модуля qr.h
Решение системы линейных уравнений  $Ax = b$ 
с помощью QR-разложения,

$$A = \begin{pmatrix} 0.1 & 0.5 & 0.6 \\ 0.2 & 0.7 & 0.9 \\ 0.3 & 1.1 & 1.3 \end{pmatrix}, \quad b = \begin{pmatrix} 1.2 \\ 1.8 \\ 2.7 \end{pmatrix}.$$

*/

#include "nl.h"

int main()
{
    double **A, *b, *t;
    nl_index n = 3;
```

```
A = nl_dmatrix_create(n, n);
b = nl_dvector_create(n);
t = nl_dvector_create(n);

A[0][0] = .1; A[0][1] = .5; A[0][2] = .6;   b[0] = 1.2;
A[1][0] = .2; A[1][1] = .7; A[1][2] = .9;   b[1] = 1.8;
A[2][0] = .3; A[2][1] = 1.1; A[2][2] = 1.3; b[2] = 2.7;

printf("Матрица A:\n");
nl_dmatrix_print(A, n, n, NULL);

qr_decomp(A, n, n, t);

printf("\nQR-разложение:\n");
nl_dmatrix_print(A, n, n, NULL);

printf("\nВектор t:\n");
nl_dvector_print(t, n, NULL);

printf("\nВектор b:\n");
nl_dvector_print(b, n, NULL);

qr_solve (A, n, t, b);

printf("\nРешение системы Ax = b:\n");
nl_dvector_print(b, n, NULL);

nl_dmatrix_free(A, n);
nl_dvector_free(b);
nl_dvector_free(t);

return 0;
}
```

Результаты работы программы:

Матрица A:

0.100	0.500	0.600
0.200	0.700	0.900
0.300	1.100	1.300

QR-разложение:

-0.374	-1.390	-1.684
0.422	0.136	0.147
0.633	0.410	0.059

Вектор t:

1.267	1.712	2.000
-------	-------	-------

Вектор b:

1.200	1.800	2.700
-------	-------	-------

Решение системы $Ax = b$:

1.000	1.000	1.000
-------	-------	-------

Для решения задачи наименьших квадратов необходимо воспользоваться функцией *qr_least_squares*. Приведем пример.

Листинг xqrls.c

/*

Пример использования функций из модуля *qr.h*

Решение задачи наименьших квадратов с помощью

QR-разложения

$$A = \begin{pmatrix} 1 & 5 & 1 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 2 \end{pmatrix}$$

*/

#include "nl.h"

int main()

{

double **A, *b, *t, *r;

nl_index m = 4;

nl_index n = 3;

A = nl_dmatrix_create(m, n);

b = nl_dvector_create(m);

t = nl_dvector_create(n);

r = nl_dvector_create(m);

A[0][0] = 1; A[0][1] = 5; A[0][2] = 1; b[0] = 1;

A[1][0] = 2; A[1][1] = 6; A[1][2] = 10; b[1] = 1;

A[2][0] = 3; A[2][1] = 7; A[2][2] = 11; b[2] = 1;

A[3][0] = 4; A[3][1] = 8; A[3][2] = 12; b[3] = 2;

printf("Матрица A:\n");

nl_dmatrix_print(A, m, n, NULL);

qr_decomp(A, m, n, t);

printf("\nQR-разложение:\n");

nl_dmatrix_print(A, m, n, NULL);

printf("\nВектор t:\n");

nl_dvector_print(t, n, NULL);

printf("\nВектор b:\n");

nl_dvector_print(b, m, NULL);

```

qr_least_squares (A, m, n, t, b, r);

printf("\nПсевдорешение системы Ax = b:\n");
nl_dvector_print(b, n, NULL);

printf("\nНевязки:\n");
nl_dvector_print(r, m, NULL);

nl_dmatrix_free(A, m);
nl_dvector_free(b);
nl_dvector_free(t);
nl_dvector_free(r);

return 0;
}

```

Результаты работы программы:

Матрица A:

1.000	5.000	1.000
2.000	6.000	10.000
3.000	7.000	11.000
4.000	8.000	12.000

QR-разложение:

-5.477	-12.780	-18.623
0.309	-3.266	0.000
0.463	-0.327	-4.382
0.618	-0.789	0.395

Вектор t:

1.183	1.156	1.730
-------	-------	-------

Вектор b :

1.000 1.000 1.000 2.000

Псевдорешение системы $Ax = b$:

0.458 0.125 -0.083

Невязки:

-0.000 -0.167 0.333 -0.167

Подробное описание используемых функций см. в документации [33].

3.6. Сингулярное разложение (SVD)

Рассмотрим матрицу A размера $m \times n$, где $m \geq n$. Можно доказать, что существуют матрицы U размера $m \times n$, V размера $n \times n$ и Σ размера $n \times n$, такие, что

$$A = U\Sigma V^T, \quad (42)$$

$U^T U = E$, $V^T V = E$ и $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$, где $\sigma_1 \geq \dots \geq \sigma_n \geq 0$. Разложение (42) называется *сингулярным разложением*³ (SVD). Столбцы u_1, \dots, u_n матрицы U называются *левыми сингулярными векторами*, столбцы v_1, \dots, v_n матрицы V — *правыми сингулярными векторами*, величины $\sigma_1, \dots, \sigma_n$ — *сингулярными числами*. Заметим, что каждая из двух систем: левых сингулярных векторов и правых сингулярных векторов — ортонормирована. Из (42) следует $A = \sum_{i=1}^n \sigma_i u_i v_i^T$.

³Иногда *сингулярным разложением* называют представление матрицы A в виде произведения $A = \tilde{U} \tilde{\Sigma} \tilde{V}^T$, в котором обе матрицы \tilde{U} и \tilde{V} — квадратные ортогональные порядка m и n соответственно. Чтобы такое разложение получить по разложению (42), достаточно столбцы матрицы U дополнить до ортонормированной системы m векторов, из построенных столбцов составить матрицу S и положить $\tilde{U} = (U, S)$, $\tilde{\Sigma} = \Sigma$, $\tilde{V} = V$. Обратный переход осуществляется отбрасыванием в \tilde{U} последних $m - n$ столбцов.

При $m < n$ необходимо рассматривать сингулярное разложение матрицы A^T .

Нетрудно доказать следующие важные свойства сингулярного разложения:

- 1) Справедливо $\text{cond}_2 A = \sigma_1/\sigma_n$. Если матрица A — квадратная, то $\|A\|_2 = \sigma_1$. Если A — квадратная и невырожденная, то $\|A^{-1}\|_2 = 1/\sigma_n$.
- 2) Если $\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0$, то $\text{rank } A = r$.
- 3) $\lambda_j(A^T A) = \sigma_j^2$, причем столбцы матрицы V образуют ортонормированную систему из собственных векторов матрицы $A^T A$.
- 4) Если $\text{rank } A = n$, то (единственное) псевдорешение системы $Ax = b$ равно $x = V\Sigma^{-1}U^T b$.

Докажем свойство 4). Так как $U^T U = E$, то столбцы матрицы U можно дополнить до ортонормированной системы m векторов⁴. Из построенных столбцов составим матрицу S . Имеем $S^T U = 0$. Матрица (U, S) — ортогональная, т.е. $(U, S)^{-1} = (U, S)^T$, поэтому для любого вектора x имеем $\|(U, S)^T x\|_2 = \|x\|_2$. Теперь получаем:

$$\begin{aligned} \|Ax - b\|_2^2 &= \|(U, S)^T (U\Sigma V^T x - b)\|_2^2 = \left\| \begin{pmatrix} \Sigma V^T x - U^T b \\ -S^T b \end{pmatrix} \right\|_2^2 = \\ &= \|\Sigma V^T x - U^T b\|_2^2 + \|S^T b\|_2^2. \end{aligned}$$

Минимум достигается тогда и только тогда, когда первое слагаемое равно нулю, т.е. при $x = V\Sigma^{-1}U^T b$. Заметим, что при этом

$$\min_x \|Ax - b\|_2 = \|S^T b\|_2.$$

⁴Ср. предыдущее примечание.

Величина

$$\text{cond}_{\text{LS}}(A, b) = \frac{2 \text{cond}_2 A}{\cos \theta} + \text{tg } \theta \cdot \text{cond}_2^2 A$$

называется *числом обусловленности для задачи наименьших квадратов*. Получаем, что

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leq \varepsilon \text{cond}_{\text{LS}}(A, b) + O(\varepsilon^2).$$

Итак, $\text{cond}_{\text{LS}}(A, b)$ является мерой чувствительности задачи наименьших квадратов, показывающей, как сильно решение этой задачи зависит от возмущения входных данных.

Формулу (35) можно интерпретировать следующим образом. Если угол θ мал, то $\text{cond}_{\text{LS}}(A, b) \approx 2 \text{cond}_2 A$, т. е. задача наименьших квадратов обусловлена примерно так же, как и задача решения систем линейных уравнений.

Если угол θ не мал, но и не близок к $\pi/2$, то $\text{cond}_{\text{LS}}(A, b) \approx \text{cond}_2^2 A$.

Если $\theta \approx \pi/2$, т. е. задача близка к вырожденной, то величина $\text{cond}_{\text{LS}}(A, b)$ растет неограничено, даже если $\text{cond}_2^2 A$ имеет умеренные значения.

3.3. Нормальные уравнения

Покажем, что множество псевдорешений системы $Ax = b$ совпадает с множеством решений системы

$$A^T Ax = A^T b, \quad (36)$$

называемой системой *нормальных уравнений*.

Имеем

$$\begin{aligned} \|A(x + \Delta x) - b\|_2^2 &= (A(x + \Delta x) - b)^T (A(x + \Delta x) - b) = \\ &= (A\Delta x)^T (A\Delta x) + (Ax - b)^T (Ax - b) + 2(A\Delta x)^T (Ax - b) = \\ &= \|A\Delta x\|_2^2 + \|Ax - b\|_2^2 + 2\Delta x^T (A^T Ax - A^T b). \end{aligned}$$

Если x — псевдорешение, то $\|A\Delta x\|_2^2 + 2\Delta x^T(A^T Ax - A^T b) \geq 0$ для любого Δx , откуда $A^T Ax - A^T b = 0$. Обратно, если $A^T Ax = A^T b$, то $\|A(x + \Delta x) - b\|_2^2 \geq \|Ax - b\|_2^2$ для любого Δx , т. е. x — псевдорешение.

Заметим, что матрица $A^T A$ — квадратная и неотрицательно определенная. Система (36) всегда совместна. Она имеет единственное решение тогда и только тогда, когда псевдорешение исходной системы единственно. Последнее имеет место в том и только в том случае, когда столбцы матрицы A линейно независимы (т. е. $\text{rank } A = n$). В этом (и только в этом) случае матрица $A^T A$ положительно определена и для решения системы (36) мы можем применить разложение Холецкого. Тогда на вычисление матричного произведения $A^T A$ требуется $mn^2 + O(n^2)$ операций с плавающей точкой (достаточно вычислить только элементы над диагональю), а на решение системы (36) требуется $n^3/3 + O(n^2)$ операций с плавающей точкой. Так как $m \geq n$, то основная масса времени тратится на формирование матрицы $A^T A$.

Недостатком метода нормальных уравнений является плохая обусловленность получаемой системы. Действительно, справедливо равенство $\text{cond}_2(A^T A) = \text{cond}_2^2 A$. Используя результаты анализа ошибок для метода Холецкого, приходим к выводу, что относительная ошибка получаемого решения задачи наименьших квадратов не превосходит некоторого умеренного кратного величины $\varepsilon_M \text{cond}_2^2 A$. Следует ожидать, что при решении нормальных уравнений с небольшой невязкой исходной системы будет потеряно вдвое больше разрядов, чем при использовании QR -разложения или сингулярного разложения.

Тем не менее использование нормальных уравнений заманчиво при решении линейной задачи наименьших квадратов при малых значениях n : например, при $n = 2$ система (36) имеет размеры 2×2 .

3.4. QR-разложение

Можно доказать, что для произвольной $(m \times n)$ -матрицы A ранга n существуют и единственны матрицы Q размера $m \times n$ с ортонормированными столбцами (т.е. $Q^T Q = E$) и верхнетреугольная матрица R размера $n \times n$ с положительными диагональными элементами, такие, что

$$A = QR. \quad (37)$$

Разложение (37) называется *QR-разложением*¹ матрицы A .

Сперва покажем, как решать линейную задачу наименьших квадратов $\min \|Ax - b\|_2$, если QR-разложение матрицы A известно. Методы построения самого QR-разложения будут рассмотрены позднее.

Итак, $Q^T Q = E$. Дополним столбцы матрицы Q до ортонормированной системы m векторов². Из построенных столбцов составим матрицу S . Имеем $S^T Q = 0$. Матрица (Q, S) — ортогональная, т.е. $(Q, S)^{-1} = (Q, S)^T$, поэтому для любого вектора x справедливо $\|(Q, S)^T x\|_2 = \|x\|_2$. Теперь получаем:

$$\|Ax - b\|_2^2 = \|(Q, S)^T (QRx - b)\|_2^2 = \left\| \begin{pmatrix} Q^T QRx - Q^T b \\ S^T QRx - S^T b \end{pmatrix} \right\|_2^2 =$$

¹Иногда QR-разложением называют представление матрицы A в виде произведения $A = \tilde{Q}\tilde{R}$, в котором матрица \tilde{Q} — квадратная ортогональная порядка m , а матрица \tilde{R} — верхнетреугольная размера $m \times n$. Чтобы такое разложение получить по разложению (37), достаточно столбцы матрицы Q дополнить до ортонормированной системы m векторов, из построенных столбцов составить матрицу S и положить

$$\tilde{Q} = (Q, S), \quad \tilde{R} = \begin{pmatrix} R \\ 0 \end{pmatrix}.$$

Обратный переход осуществляется отбрасыванием в \tilde{Q} последних $m - n$ столбцов, а в \tilde{R} — последних $m - n$ строк.

²Ср. предыдущее примечание.

$$= \|Rx - Q^T b\|_2^2 + \|S^T b\|_2^2.$$

Минимум достигается тогда и только тогда, когда первое слагаемое равно нулю. Итак, решением линейной задачи наименьших квадратов является решение треугольной системы

$$Rx = Q^T b,$$

при этом

$$\min_x \|Ax - b\|_2 = \|S^T b\|_2.$$

Заметим, что матрица R невырождена тогда и только тогда, когда $\text{rank } A = n$. В этом случае $x = R^{-1}Q^T b$.

Рассмотрим несколько методов построения QR -разложения.

3.4.1. Ортогонализация Грама–Шмидта

Процесс ортогонализации Грама–Шмидта — это классический метод построения QR -разложения. Он заключается в построении ортонормированного базиса q_1, \dots, q_n подпространства, натянутого на столбцы a_1, \dots, a_n , причем

$$L(q_1, \dots, q_j) = L(a_1, \dots, a_j) \quad (j = 1, 2, \dots, n). \quad (38)$$

Матрица, составленная из столбцов q_1, \dots, q_n , — это матрица Q . Матрица перехода от базиса q_1, \dots, q_n к a_1, \dots, a_n — это матрица R , т. е. $a_j = \sum_{i=1}^j r_{ij} q_i$

Опишем этот алгоритм подробнее.

for $j = 1, \dots, n$

$q_j = a_j$

for $i = 1, \dots, j - 1$

$r_{ij} = q_i^T a_j$

$q_j = q_j - r_{ij} q_i$

end

$r_{jj} = \|q_j\|_2$

(*)

```

if  $r_{jj} = 0$ 
    exit ( $a_i$  линейно зависит от  $a_1, \dots, a_{i-1}$ )
end
 $q_j = q_j / r_{jj}$ 
end

```

Алгоритм использует $2mn^2 + O(mn)$ операций с плавающей точкой.

Заметим, что присваивание $r_{ij} = q_i^T a_j$ в строке (*) приведенного алгоритма можно заменить на $r_{ij} = q_i^T q_j$. Полученный таким образом новый алгоритм (в точной арифметике эквивалентный исходному) называется *модифицированным процессом Грама–Шмидта*.

При реальных вычислениях на компьютере (при наличии ошибок округления) процесс ортогонализации Грама–Шмидта (в том числе модифицированный) вместо векторов q_1, q_2, \dots, q_n возвратит векторы $q_1, q_2 + \Delta q_2, \dots, q_n + \Delta q_n$. Например, в режиме накопления

$$L(q_1, q_2 + \Delta q_2, \dots, q_j + \Delta q_j) = L(a_1, a_2 + \Delta a_2, \dots, a_j + \Delta a_j),$$

где $\|\Delta a_k\|_2 \leq 2\varepsilon_M \|a_k\|_2$, т.е. линейные оболочки полученных векторов совпадают с линейными оболочками «слабо возмущенных» исходных векторов [2].

Итак, в отношении выполнения равенства (38) процесс ортогонализации Грама–Шмидта ведет себя крайне устойчиво. К сожалению, этого нельзя сказать о выполнении условия ортогональности полученной системы. В большинстве случаев процесс Грама–Шмидта приводит к значительному нарушению ортогональности. Модифицированный процесс Грама–Шмидта чуть лучше, но и он не свободен от указанного недостатка. Пусть $Q + \Delta Q$ — реально вычисленная матрица. Можно показать [4], что

$$(Q + \Delta Q)^T \cdot (Q + \Delta Q) = E + \Delta E,$$

где $\|\Delta E\|_2 \lesssim \varepsilon_M \operatorname{cond}_2 A$.

Для исправления отмеченного недостатка используют *пертортогонализацию* [2].

Тем не менее, как оказалось, ортогональность столбцов матрицы Q не играет решающей роли в задаче наименьших квадратов. Можно показать, что алгоритм решения задачи наименьших квадратов с помощью модифицированного процесса ортогонализации Грама–Шмидта обратно устойчив и, следовательно может быть использован на практике [4]. В любом случае, затраты в методах типа Грама–Шмидта чуть выше, чем в методе Хаусхолдера.

3.4.2. Отражения Хаусхолдера

Метод отражений Хаусхолдера — один из самых распространенных методов нахождения QR -разложения.

Пусть v — n -мерный ненулевой вектор-столбец. Квадратная матрица P порядка n вида

$$P = E - \frac{2vv^T}{v^T v}$$

называется *отражением Хаусхолдера* или просто *отражением*. Вектор v называется *вектором Хаусхолдера*. Умножению матрицы P на вектор x можно дать следующую геометрическую интерпретацию: вектор Px получается отражением вектора x относительно гиперплоскости, ортогональной вектору v . Легко проверить, что матрица Хаусхолдера симметрична и ортогональна.

Пусть $x \neq 0$. Найдем отражение P , такое, что $Px = \alpha e_1$ для какого-либо α . Имеем

$$Px = \left(E - \frac{2vv^T}{v^T v} \right) x = x - \frac{2v(v^T x)}{v^T v} = x - \frac{2v^T x}{v^T v} v,$$

и поэтому v коллинеарен $x - Px = x - \alpha e_1$. Так как v определен с точностью до ненулевого множителя, то можно положить $v =$

$x - \alpha e_1$. Ввиду ортогональности матрицы P имеем $\|Px\|_2 = \|x\|_2$, откуда $\alpha = \pm\|x\|_2$. Поэтому

$$v = x \pm \|x\|_2 e_1. \quad (39)$$

Итак, в качестве вектора Хаусхолдера v можно взять вектор, вычисляемый по формуле (39), или любой ненулевой, ему коллинеарный.

С алгоритмом построения вектора Хаусхолдера связаны некоторые важные детали. Одна из них — выбор знака в формуле (39). Если x почти коллинеарен e_1 , то вектор $v = x - \text{sign}(x_1)\|x\|_2 e_1$ имеет малую норму. Вследствие этого возможно появление большой относительной ошибки при вычислении множителя $2/v^T v$. Эту трудность можно обойти, взяв α с тем же знаком, что и первая компонента вектора x :

$$v = x + \text{sign}(x_1)\|x\|_2 e_1. \quad (40)$$

Это обеспечивает выполнение неравенства $\|v\|_2 \geq \|x\|_2$ и гарантирует почти точную ортогональность вычисленной матрицы P .

Другая деталь связана с выбором множителя для вектора, вычисляемого по формуле (39). Мы будем использовать такой множитель, что $v_1 = 1$. Это несколько упрощает способ хранения ортогональных матриц, вычисленных по методу отражений Хаусхолдера.

Суть алгоритма QR-разложения на основе отражений сначала поясним на примере. Пусть $m = 5, n = 4$ и матрицы P_1 и P_2 таковы, что

$$P_2 P_1 A = \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ 0 & f_{22} & f_{23} & f_{24} \\ 0 & 0 & f_{33} & f_{34} \\ 0 & 0 & f_{43} & f_{44} \\ 0 & 0 & f_{53} & f_{54} \end{pmatrix}.$$

Покажем, как на месте элементов f_{43} , f_{53} получить нули. Найдем матрицу отражения P'_3 , такую, что

$$P'_3 \times \begin{pmatrix} f_{33} \\ f_{43} \\ f_{53} \end{pmatrix} = \begin{pmatrix} f'_{33} \\ 0 \\ 0 \end{pmatrix}.$$

Если $P_3 = \text{diag}(1, 1, P'_3)$, то

$$P_3 P_2 P_1 A = \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ 0 & f_{22} & f_{23} & f_{24} \\ 0 & 0 & f'_{33} & f'_{34} \\ 0 & 0 & 0 & f'_{44} \\ 0 & 0 & 0 & f'_{54} \end{pmatrix}.$$

Выполнив n таких шагов, получим верхнюю треугольную матрицу $\tilde{R} = P_n P_{n-1} \dots P_1 A$. Откуда, так как матрицы P_j — ортогональные и симметричные, имеем $A = \tilde{Q} \tilde{R}$, где $\tilde{Q} = P_1 P_2 \dots P_n$. Пусть Q состоит из первых n столбцов матрицы \tilde{Q} , а R состоит из первых n строк матрицы \tilde{R} , тогда $A = QR$. Это и есть искомое разложение.

Матрицы P_j не требуется формировать в явном виде: достаточно хранить соответствующий вектор v_j , который можно записывать в j -й столбец матрицы A .

Для решения задачи наименьших квадратов необходимо вычислить произведение $Q^T b = P_n P_{n-1} \dots P_1 b$.

Трудоемкость алгоритма построения QR -разложения на основе отражений составляет $2n^2 m - (2/3)n^3 + O(mn)$ арифметических операций с плавающей точкой. Для получения решения задачи наименьших квадратов при известном QR -разложении требуется только $O(mn)$ арифметических операций.

Пусть при реальных вычислениях на компьютере в результате ошибок округления вместо Q получена матрица $Q + \Delta Q$.

Можно показать [4], что

$$(Q + \Delta Q)^T \cdot (Q + \Delta Q) = E + \Delta E,$$

где $\|\Delta E\|_2 \lesssim \varepsilon_M$.

Можно показать [13], что при наличии ошибок округления вычисленное по методу Хаусхолдера решение $x + \Delta x$ задачи $\min \|Ax - b\|_2$ является точным решением «близкой» задачи

$$\min \|(A + \Delta A)x - (b + \Delta b)\|_2,$$

где

$$\begin{aligned} \|\Delta A\|_F &\leq (6m - 3n + 41)n\varepsilon_M \|A\|_F + O(\varepsilon_M^2), \\ \|\Delta b\|_2 &\leq (6m - 3n + 40)n\varepsilon_M \|b\|_2 + O(\varepsilon_M^2). \end{aligned} \quad (41)$$

Как мы видим, относительная ошибка эквивалентного возмущения пропорциональна умеренному кратному числу ε_M . Используя теорию возмущений из раздела 3.2, получаем, что при малой невязке относительная ошибка решения задачи наименьших квадратов примерно пропорциональна $\varepsilon_M \operatorname{cond}_2 A$, а при большой невязке — $\varepsilon_M \operatorname{cond}_2^2 A$.

На выходе диагональ и верхняя треугольная часть матрицы A содержат элементы матрицы R . Матрица Q представлена элементами ниже диагонали и вектором t . Вектор t имеет длину n . Матрица Q определяется произведением матриц отражений Хаусхолдера.

3.4.3. Вращения Гивенса

Вращением Гивенса, или просто *вращением*, называется матрица

$$R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}.$$

Умножению матрицы R на вектор x можно дать следующую геометрическую интерпретацию: вектор Rx получается вращением вектора x на угол θ . Вращение n -мерного пространства в плоскости x_i, x_j определяет матрица

$$R = \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & \cos \theta & & & -\sin \theta & \\ & & & & 1 & & & \\ & & & & & \ddots & & \\ & & & & & & 1 & \\ & & \sin \theta & & & & \cos \theta & \\ & & & & & & & 1 \\ & & & & & & & & \ddots & \\ & & & & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ j \\ \\ \\ i \\ \\ \\ \\ \end{matrix}$$

Очевидно, что матрица вращения ортогональна.

Если

$$\cos \theta = \frac{x_j}{\sqrt{x_i^2 + x_j^2}}, \quad \sin \theta = -\frac{x_i}{\sqrt{x_i^2 + x_j^2}},$$

то

$$\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \times \begin{pmatrix} x_j \\ x_i \end{pmatrix} = \begin{pmatrix} \sqrt{x_i^2 + x_j^2} \\ 0 \end{pmatrix}.$$

Таким образом, с помощью домножения слева вектора x на матрицу вращения R можно занулить одну (i -ю) компоненту. Заметим, что при этом все компоненты, кроме i -й и j -й, не изменяются.

Алгоритм QR -разложения на основе вращений аналогичен алгоритму, использующему отражения, с тем отличием, что

элементы матрицы A при обработке j -го столбца в нем зануляются по одному (для определенности, например, сверху вниз).

Чтобы перейти, например, от матрицы

$$\begin{pmatrix} f_{11} & f_{12} & f_{13} \\ 0 & f_{22} & f_{23} \\ 0 & f_{32} & f_{33} \\ 0 & f_{42} & f_{43} \end{pmatrix} \quad \text{к матрице} \quad \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ 0 & g_{22} & g_{23} \\ 0 & 0 & g_{33} \\ 0 & 0 & g_{43} \end{pmatrix},$$

производим умножения:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & -s & 0 \\ 0 & s & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ 0 & f_{22} & f_{23} \\ 0 & f_{32} & f_{33} \\ 0 & f_{42} & f_{43} \end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ 0 & f'_{22} & f'_{23} \\ 0 & 0 & f'_{33} \\ 0 & f_{42} & f_{43} \end{pmatrix},$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c' & 0 & -s' \\ 0 & 0 & 1 & 0 \\ 0 & s' & 0 & c' \end{pmatrix} \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ 0 & f'_{22} & f'_{23} \\ 0 & 0 & f'_{33} \\ 0 & f_{42} & f_{43} \end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ 0 & f''_{22} & f''_{23} \\ 0 & 0 & f'_{33} \\ 0 & 0 & f''_{43} \end{pmatrix}.$$

Для хранения элементов найденного разложения дополнительная память не нужна. Информация о матрице вращения, с помощью которой мы занулили соответствующий элемент, хранится на его месте.

Для этого применяют следующий метод [5]. Пусть $s = \sin \theta$, $c = \cos \theta$. Если $|s| < |c|$, то в память записывается число $s \cdot \text{sign}(c)$, в противном случае записывается $\text{sign}(s)/c$.

По хранимому значению p восстановление чисел s и c осуществляется следующим образом. Если $|p| < 1$, то $s = p$ и $c = \sqrt{1 - s^2}$, в противном случае $c = 1/p$ и $s = \sqrt{1 - c^2}$. Заметим, что вместо исходных s , c мы можем восстановить $-s$, $-c$, но для наших целей приемлема и такая пара.

Метод вращений в два полтора раза медленнее метода отражений. Существует метод *быстрых вращений* [4], который, все-таки, медленнее метода отражений.

Обратный анализ ошибок для метода Гивенса решения задачи наименьших квадратов приводит к оценкам, аналогичным (41) [4]. Относительная ошибка эквивалентного возмущения пропорциональна умеренному кратному числа ε_M .

3.5. QR-разложение в библиотеке NL

Метод QR-разложения на основе отражений Хаусхолдера реализован в функции *qr_decomp*. После того как разложение найдено, для решения квадратной совместной системы линейных уравнений необходимо воспользоваться функцией *qr_solve*. Приведем пример.

Листинг xqr.c

```
/*
Пример использования функций из модуля qr.h
Решение системы линейных уравнений  $Ax = b$ 
с помощью QR-разложения,

$$A = \begin{pmatrix} 0.1 & 0.5 & 0.6 \\ 0.2 & 0.7 & 0.9 \\ 0.3 & 1.1 & 1.3 \end{pmatrix}, \quad b = \begin{pmatrix} 1.2 \\ 1.8 \\ 2.7 \end{pmatrix}.$$

*/

#include "nl.h"

int main()
{
    double **A, *b, *t;
    nl_index n = 3;
```

```
A = nl_dmatrix_create(n, n);
b = nl_dvector_create(n);
t = nl_dvector_create(n);

A[0][0] = .1; A[0][1] = .5; A[0][2] = .6;   b[0] = 1.2;
A[1][0] = .2; A[1][1] = .7; A[1][2] = .9;   b[1] = 1.8;
A[2][0] = .3; A[2][1] = 1.1; A[2][2] = 1.3; b[2] = 2.7;

printf("Матрица A:\n");
nl_dmatrix_print(A, n, n, NULL);

qr_decomp(A, n, n, t);

printf("\nQR-разложение:\n");
nl_dmatrix_print(A, n, n, NULL);

printf("\nВектор t:\n");
nl_dvector_print(t, n, NULL);

printf("\nВектор b:\n");
nl_dvector_print(b, n, NULL);

qr_solve (A, n, t, b);

printf("\nРешение системы Ax = b:\n");
nl_dvector_print(b, n, NULL);

nl_dmatrix_free(A, n);
nl_dvector_free(b);
nl_dvector_free(t);

return 0;
}
```

Результаты работы программы:

Матрица A:

0.100	0.500	0.600
0.200	0.700	0.900
0.300	1.100	1.300

QR-разложение:

-0.374	-1.390	-1.684
0.422	0.136	0.147
0.633	0.410	0.059

Вектор t:

1.267	1.712	2.000
-------	-------	-------

Вектор b:

1.200	1.800	2.700
-------	-------	-------

Решение системы $Ax = b$:

1.000	1.000	1.000
-------	-------	-------

Для решения задачи наименьших квадратов необходимо воспользоваться функцией *qr_least_squares*. Приведем пример.

Листинг xqrls.c

/*

Пример использования функций из модуля *qr.h*

Решение задачи наименьших квадратов с помощью

QR-разложения

$$A = \begin{pmatrix} 1 & 5 & 1 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 2 \end{pmatrix}$$

*/

#include "nl.h"

int main()

{

double **A, *b, *t, *r;

nl_index m = 4;

nl_index n = 3;

A = nl_dmatrix_create(m, n);

b = nl_dvector_create(m);

t = nl_dvector_create(n);

r = nl_dvector_create(m);

A[0][0] = 1; A[0][1] = 5; A[0][2] = 1; b[0] = 1;

A[1][0] = 2; A[1][1] = 6; A[1][2] = 10; b[1] = 1;

A[2][0] = 3; A[2][1] = 7; A[2][2] = 11; b[2] = 1;

A[3][0] = 4; A[3][1] = 8; A[3][2] = 12; b[3] = 2;

printf("Матрица A:\n");

nl_dmatrix_print(A, m, n, NULL);

qr_decomp(A, m, n, t);

printf("\nQR-разложение:\n");

nl_dmatrix_print(A, m, n, NULL);

printf("\nВектор t:\n");

nl_dvector_print(t, n, NULL);

printf("\nВектор b:\n");

nl_dvector_print(b, m, NULL);

```

qr_least_squares (A, m, n, t, b, r);

printf("\nПсевдорешение системы Ax = b:\n");
nl_dvector_print(b, n, NULL);

printf("\nНевязки:\n");
nl_dvector_print(r, m, NULL);

nl_dmatrix_free(A, m);
nl_dvector_free(b);
nl_dvector_free(t);
nl_dvector_free(r);

return 0;
}

```

Результаты работы программы:

Матрица A:

1.000	5.000	1.000
2.000	6.000	10.000
3.000	7.000	11.000
4.000	8.000	12.000

QR-разложение:

-5.477	-12.780	-18.623
0.309	-3.266	0.000
0.463	-0.327	-4.382
0.618	-0.789	0.395

Вектор t:

1.183	1.156	1.730
-------	-------	-------

Вектор b :

1.000 1.000 1.000 2.000

Псевдорешение системы $Ax = b$:

0.458 0.125 -0.083

Невязки:

-0.000 -0.167 0.333 -0.167

Подробное описание используемых функций см. в документации [33].

3.6. Сингулярное разложение (SVD)

Рассмотрим матрицу A размера $m \times n$, где $m \geq n$. Можно доказать, что существуют матрицы U размера $m \times n$, V размера $n \times n$ и Σ размера $n \times n$, такие, что

$$A = U\Sigma V^T, \quad (42)$$

$U^T U = E$, $V^T V = E$ и $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$, где $\sigma_1 \geq \dots \geq \sigma_n \geq 0$. Разложение (42) называется *сингулярным разложением*³ (SVD). Столбцы u_1, \dots, u_n матрицы U называются *левыми сингулярными векторами*, столбцы v_1, \dots, v_n матрицы V — *правыми сингулярными векторами*, величины $\sigma_1, \dots, \sigma_n$ — *сингулярными числами*. Заметим, что каждая из двух систем: левых сингулярных векторов и правых сингулярных векторов — ортонормирована. Из (42) следует $A = \sum_{i=1}^n \sigma_i u_i v_i^T$.

³Иногда *сингулярным разложением* называют представление матрицы A в виде произведения $A = \tilde{U} \tilde{\Sigma} \tilde{V}^T$, в котором обе матрицы \tilde{U} и \tilde{V} — квадратные ортогональные порядка m и n соответственно. Чтобы такое разложение получить по разложению (42), достаточно столбцы матрицы U дополнить до ортонормированной системы m векторов, из построенных столбцов составить матрицу S и положить $\tilde{U} = (U, S)$, $\tilde{\Sigma} = \Sigma$, $\tilde{V} = V$. Обратный переход осуществляется отбрасыванием в \tilde{U} последних $m - n$ столбцов.

При $m < n$ необходимо рассматривать сингулярное разложение матрицы A^T .

Нетрудно доказать следующие важные свойства сингулярного разложения:

- 1) Справедливо $\text{cond}_2 A = \sigma_1/\sigma_n$. Если матрица A — квадратная, то $\|A\|_2 = \sigma_1$. Если A — квадратная и невырожденная, то $\|A^{-1}\|_2 = 1/\sigma_n$.
- 2) Если $\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0$, то $\text{rank } A = r$.
- 3) $\lambda_j(A^T A) = \sigma_j^2$, причем столбцы матрицы V образуют ортонормированную систему из собственных векторов матрицы $A^T A$.
- 4) Если $\text{rank } A = n$, то (единственное) псевдорешение системы $Ax = b$ равно $x = V\Sigma^{-1}U^T b$.

Докажем свойство 4). Так как $U^T U = E$, то столбцы матрицы U можно дополнить до ортонормированной системы m векторов⁴. Из построенных столбцов составим матрицу S . Имеем $S^T U = 0$. Матрица (U, S) — ортогональная, т. е. $(U, S)^{-1} = (U, S)^T$, поэтому для любого вектора x имеем $\|(U, S)^T x\|_2 = \|x\|_2$. Теперь получаем:

$$\begin{aligned} \|Ax - b\|_2^2 &= \|(U, S)^T (U\Sigma V^T x - b)\|_2^2 = \left\| \begin{pmatrix} \Sigma V^T x - U^T b \\ -S^T b \end{pmatrix} \right\|_2^2 = \\ &= \|\Sigma V^T x - U^T b\|_2^2 + \|S^T b\|_2^2. \end{aligned}$$

Минимум достигается тогда и только тогда, когда первое слагаемое равно нулю, т. е. при $x = V\Sigma^{-1}U^T b$. Заметим, что при этом

$$\min_x \|Ax - b\|_2 = \|S^T b\|_2.$$

⁴Ср. предыдущее примечание.

Литература

Числа в квадратных скобках, приведенные в конце описания каждого источника, указывают на страницы настоящего издания, на которых имеются соответствующие ссылки. Начиная с номера [24], приводятся Интернет-адреса библиотек численных методов.

1. *Богачев К. Ю.* Практикум на ЭВМ. Методы решения линейных систем и нахождения собственных значений. — М.: Изд-во механико-математического ф-та Моск. ун-та, 1998. [8]
2. *Воеводин В. В.* Вычислительные основы линейной алгебры. — М.: Наука, 1977. [8, 17, 33, 42, 71, 72]
3. *Воеводин В. В., Кузнецов Ю. А.* Матрицы и вычисления. — М.: Наука, 1984. [8]
4. *Голуб Дж., Ван Лоун Ч.* Матричные вычисления. — М.: Мир, 1999. [8, 33, 52, 66, 71, 72, 75, 78, 86, 87, 88, 118, 122, 127, 129, 130]
5. *Деммель Дж.* Вычислительная линейная алгебра. Теория и приложения. — М.: Мир, 2001. [8, 17, 19, 33, 49, 50, 51, 52, 53, 66, 77, 87, 118, 119, 121, 122, 123, 127, 129]
6. *Джордж А., Лю Дж.* Численное решение больших разреженных систем уравнений. — М.: Мир, 1984. [94, 105, 174, 175]
7. *Златев З., Эстербю О.* Прямые методы для разреженных матриц. — М.: Мир, 1987. [94]
8. *Икрамов Х. Д.* Численные методы для симметричных линейных систем. — М.: Наука, 1988. [61]
9. *Ильин В. П.* Методы неполной факторизации для решения линейных систем. — М.: Физматлит, 1995. [176]
10. *Каханер Д., Моулер К., Нэш С.* Численные методы и программное обеспечение. — М.: Мир, 1999. [8, 50, 52]

11. *Кетков Ю.Л., Кетков А.Ю., Шульц М.М.* МАТЛАВ 6.x.: программирование численных методов. — СПб.: БХВ-Петербург, 2004. [142]
12. *Кнут Д.* Искусство программирования для ЭВМ. Том 1: Основные алгоритмы. — М.: Мир, 1973. [95]
13. *Лоусон Ч., Хенсон Р.* Численное решение задач метода наименьших квадратов. — М.: Наука, 1986. [66, 75, 86]
14. *Мартынов Н.Н.* Введение в МАТЛАВ 6.x. — М.: Кудиц-образ, 2002. [142]
15. *Писсанецки С.* Технология разреженных матриц. — М.: Мир, 1988. [94, 99, 103, 105, 174, 175]
16. *Самарский А.А., Гулин А.В.* Численные методы. — М.: Наука, 1989. [118, 120, 140]
17. *Форсайт Дж., Малькольм М., Моулер К.* Машинные методы математических вычислений. — М.: Мир, 1980. [8, 90]
18. *Anderson E., Bai Z., Bischof C., Blackford S., Demmel J., Dongarra J., Du Croz J., Greenbaum A., Hammarling S., McKenney A., Sorensen D.* LAPACK Users' Guide. Third Edition. — Philadelphia: SIAM, 1999. [8, 184]
19. *Barrett R., Berry M., Chan T.F., Demmel J., Donato J., Dongarra J., Eijkhout V., Pozo R., Romine C., Van der Vorst H.* Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. — Philadelphia: SIAM, 1994. [118, 127, 129, 134, 136]
20. IEEE Standard for Binary Floating-point Arithmetic: ANSY/IEEE std 754-1985. — 1985. Опубликован также в SIGPLAN Notices. — 1987. — 22 (2). — P. 9–25. [22]
21. *Heath M. T.* Scientific Computing, An Introductory Survey. — New York: McGraw-Hill, 2002. [8, 252]
22. *Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P.* Numerical Recipes in C. The Art of Scientific Computing. — Cambridge: Cambridge University Press, 1992. [8]

23. Using MATLAB. Version 7.0. — The MathWorks Inc., 2004. [8, 142]
24. Библиотека численного анализа НИВЦ МГУ
http://www.srcc.msu.su/num_anal/lib_na/libnal.htm [7, 257]
25. Automatically Tuned Linear Algebra Software (ATLAS)
<http://math-atlas.sourceforge.net/> [185]
26. BLAS (Basic Linear Algebra Subprograms)
<http://www.netlib.org/blas/> [185]
27. CLAPACK (f2c'ed version of LAPACK)
<http://www.netlib.org/clapack/> [230]
28. GSL — GNU Scientific Library
<http://www.gnu.org/software/gsl/> [7]
29. Intel Math Kernel Library
<http://www.intel.com/software/products/mkl/> [185]
30. LAPACK — Linear Algebra PACKage
<http://www.netlib.org/lapack/> [184, 228]
31. NAG Numerical Libraries <http://www.nag.com/> [7]
32. Netlib Repository at UTK and ORNL
<http://www.netlib.org/> [7]
33. NL. Библиотека численных методов линейной алгебры
<http://www.uic.nnov.ru/~zny/nl/> [28, 29, 31, 57, 63, 83, 93, 100, 109, 112, 117, 126, 133, 139]

Предметный указатель

ATLAS, 185

band.h, 29

band_decomp, 114

band_solve, 114

band_tridiag, 111

BERR, 19, 198

BLAS, 7, 185

chol.h, 29

chol_decomp, 61

chol_solve, 61

CLAPACK, 7, 230

CR(C)O, 98, 169

CR(C)U, 98

DGBSV, 210

DGELS, 216

DGELSS, 223

DGESV, 188, 235

dgesv_, 235

DGESVX, 192

DGTSV, 206

DPOSV, 202

EISPACK, 7, 141, 184

FERR, 18, 197

gallery.h, 29

GSL, 7

IEEE 754, 19, 20, 22, 242

LAPACK, 7, 141, 184, 228

LDL^T -разложение, 57, 105

исключение по столбцу (*kij*), 57

компактная схема (*jik*), 60

LINPACK, 7, 141, 184

lu.h, 29

LU-разложение, 36, 44, 159, 188, 192

kji-, *ikj*-, *jki*- варианты, 41

исключение по столбцу (*kij*), 38

исключение по строке (*ijk*), 39

компактная схема (*jik*), 40

неполное, 135, 176

стратегия полного выбора, 44

стратегия частичного выбора,

43

lu_decomp, 54, 114

lu_solve, 54

MATLAB, 7, 141

MKL, 185

NAG, 7

NL, 7, 28, 54, 61, 78, 90, 99, 106, 111,

114, 123, 131, 136

nl.h, 29

QL-разложение, 217

qr.h, 29

QR-разложение, 69, 161, 217

qr_least_squares, 80

qr_decomp, 78

qr_solve, 78

qsort.h, 29

RR(C)O, 95

RR(C)U, 95

RR(U)O, 98

RR(U)U, 99

SOR, см. метод релаксации

- sparse.h*, 29
- sp_chol_num*, 106
- sp_chol_symb*, 106
- sp_conj*, 136
- sp_conj_sym*, 131
- sp_gauss_seidel*, 123
- SVD, см. сингулярное разложение
- svd.h*, 29
- svd_correct*, 90
- svd_decomp*, 90
- svd_least_squares*, 90

- util.h*, 29

- xband.c*, 114
- xchol.c*, 61
- xlu.c*, 54
- xqr.c*, 78
- xqrls.c*, 80
- xsparse.c*, 100
- xspbicnj.c*, 136
- xspchol.c*, 106
- xspconjs.c*, 131
- xspseid.c*, 123
- xsvd.c*, 90
- xtridiag.c*, 111

- Алгоритм обратно устойчивый, 27
- Анализ
 - ошибок
 - обратный, 27
 - прямой, 25
 - чувствительности, 26
- Апостериорная оценка, 18
- Аппроксимация, 64
- Арифметика
 - с округлением, 20
 - с отбрасыванием, 20

- База, 19

- Ведущий элемент, 35

- Вращение Гивенса, 75
 - быстрое, 78

- Главный элемент, см. ведущий элемент

- Задача наименьших квадратов, 85
 - линейная, 64
 - неполного ранга, 86
- Заполнение, 103

- Итерационное уточнение, 52

- Компьютер двоичный, 19
 - IEEE-арифметика, 22
- Круг Гершгорина, 240

- Мантисса, 19
- Масштабирование, 54
- Матрица
 - вращения Гивенса, см. вращения Гивенса
 - ленточная, 112, 210
 - компактная схема хранения, 113
 - трехдиагональная, 109
 - ортогональная, 69
 - отражения Хаусхолдера, см. отражение Хаусхолдера
 - плохо обусловленная, 18
 - псевдообратная, 89, 167
 - разреженная, 94
 - трехдиагональная, 206
- Матрица-перестановка, 43
- Машинное эpsilon, 21, 145
- Метод
 - бисопряженных градиентов, 134, 182
 - с предобуславливанием, 136
 - вращений Гивенса, 75
 - быстрых вращений, 78
- Гаусса, 33

- без выбора ведущего элемента, 33
- обратный ход, 36
- прямой ход, 34
- Гаусса–Зейделя, 122
 - с параметром релаксации, см. метод релаксации
- Жордана–Гаусса, 247
- итерационный, 118
 - нестационарный, 118
 - стационарный, 118
- Катхилл–Макки, обратный, 105, 174
- Краута–Дулитла, см. *LU*-разложение, компактная схема минимальной степени, 105, 174
- минимальных невязок, 139
- наискорейшего спуска, 140
- отражений Хаусхолдера, 72
- прогонки, 110
- простой итерации, 119
- прямой, 33, 118
- регуляризации, 249
- релаксации, 122
- сопряженных градиентов, 126, 179
 - с предобуславливанием, 128, 180
- Холецкого (квадратного корня), 60, 105
- Якоби, 121
- Множители Гаусса, 34, 35
- Наддиагональ, 113
- Невязка, 18
- Норма, 12
 - евклидова, 12
 - кольцевая, см. норма матричная
 - манхеттенская, 12
 - матричная, 13
 - манхеттенская, 13
 - спектральная, 13
 - фробениусова (евклидова), 14
 - чебышева, 13
- подчиненная, 13
- чебышева, 12
- Нормальные уравнения, 67
- Нормы согласованные, 13
- Овал Кассини, 240
- Ортогонализация Грама–Шмидта, 70
 - модифицированная, 71
- Отношение Релея, 256
- Отражение Хаусхолдера, 72
- Оценщик обусловленности, 51
 - Хэйджера, 51, 166
- Оценщик Хэйджера, 244
- Ошибка
 - абсолютная, 17, 25
 - обратная, 27
 - покомпонентная, см. BERR
 - относительная, 17, 25
- Переортогонализация, 72
- Переупорядочение, 104
 - оптимальное, 105
- Поддиагональ, 113
- Подстановка
 - обратная, 36, 105
 - прямая, 36, 105
- Показатель, 19
- Портрет матрицы, 96
- Потеря значимости, 20
- Предобуславливание, 128, 135, 180
- Пространство нормированное, 12
- Псевдорешение, 64
 - нормальное, 87
- Разложение
 - LDL^T , см. LDL^T -разложение
 - LU , см. LU -разложение

- QL , см. QL -разложение
 QR , см. QR -разложение
символическое, 105
сингулярное, см. сингулярное разложение
Холецкого, 57, 59, 68, 105, 160, 202
исключение по столбцу (kij) , 59
неполное, 130, 178
численное, 105
- Разряд, 19
- Регуляризованная система, 249
- Режим накопления, 42
- Сингулярное разложение, 83, 87, 163, 223
- Система линейных уравнений
недоопределенная, 254
определенная, 33
переопределенная, 64
- Спектральный радиус, 119, 240
- Сходимость
линейная, 119
сверхлинейная, 131
со скоростью геометрической прогрессии, см. сходимость линейная
- Теория возмущений, см. анализ чувствительности
- Уравновешивание, см. масштабирование
- Факторизация, см. разложение
неполная, 130, 135, 176
полная, 176
- Флоп, 28
- Число
двойной точности, 25
- ненормализованное, 20
нормализованное, 20
обусловленности, 14, 26, 165
абсолютное, 26
для задачи наименьших квадратов, 67
относительное, 26
относительное покомпонентное, 19
спектральное, 66
одинарной точности, 22
с плавающей точкой, 19
сингулярное, 83
- Эквивалентное возмущение, 27

*Сергей Александрович Белов,
Николай Юрьевич Золотых*

Численные методы линейной алгебры

Лабораторный практикум

Формат $60 \times 84 \frac{1}{16}$. Бумага офсетная. Печать офсетная.
Уч.-изд. л. 11,9. Усл. печ. л. 13,6. Тираж 300 экз. Заказ

Издательство Нижегородского госуниверситета
им. Н.И. Лобачевского,
603950, Н. Новгород, пр. Гагарина, 23.

Типография ННГУ, 603000, Н. Новгород, ул. Б. Покровская, 37.
Лицензия ПД № 18–0099 от 04.05.2001.